

DRUNK: Dusting Robot Using Nice Kinematics

Alex Gaal

Austin Tang

Chenyang Wang

Haochen Yin

University of California, Los Angeles

ABSTRACT

This is the final project report for the course MAE 263A Kinematics of Robotic Systems. In this project, we designed a robotic manipulator operated in 3D with 4 DoF using Dynamixel MX-28AR servo motors as actuators that can perform surface-sweeping/wiping on small surfaces that are hard to reach by hand. The whole process can be summarized as following: we first prototyped our manipulator in CAD software and performed static force analysis; then we derived the forward and inverse kinematics and studied the workspace of our manipulator; finally we modeled the manipulator in MATLAB using the kinematics we derived and performed a simple task simulation to verify the kinematics. By modeling in MATLAB, we successfully verified our forward and inverse kinematics by showing that our end effector correctly follows our predetermined trajectory. The original plan was to build our manipulator using 3D printing and test it in a real environment, but due to COVID-19, these tasks will be conducted upon campus reopening. Detailed discussion on each step will be provided in the following sections.

1 Introduction

The use of robotic arms has become more and more popular for many applications. The most important reason for this growth can be concluded to be the low cost and high abilities of these robots. Currently, human labor is becoming more and more expensive, and compared to human labor, the cost for an entire robot system is relatively low. Therefore, in manufacturing industries, people tend to replace human workers with robots, or robotic arms in general. Besides the economic concerns, with the improvement of technology, these robots are becoming more and more capable and are advanced enough to do tasks that are dangerous or even impossible for human workers to perform. With this in mind, in this report, we will present our design of a robotic manipulator that is operated in 3D space with 4 degrees of freedom. Dynamixel MX-28AR servo motors are used as the actuators for the prismatic and revolute joints of our robot. The goal for our robot is to perform surface-sweeping/wiping on small surfaces that are hard to reach by hand. For example, our robot can be mounted on the wall above a bookshelf so that it can clean the top of the bookshelf, an area that is very hard for a human to reach while conducting everyday cleaning.

To begin with, we will first prototype our robot using the professional modeling software CATIA. With the model built in CATIA, we will then perform a static force analysis to figure out the best parameters for our robot, such as the length, thickness, and material of our robotic arms, to maximize performance given the torque constraint provided by the motors. Next, we will derive the Forward Kinematics for our robot. In general, Forward Kinematics is the static geometrical problem of calculating the position and orientation of the end effector of the manipulator given the values of the joint variables and without regard to any forces. In Forward Kinematics, some smart techniques are used to create the transformation matrices and thus compute the position and orientation of the end effector with regard to the base frame. After we obtain the transformation matrices, we then derive the inverse kinematics. The basic idea of inverse kinematics is that suppose we want to reach some desired positions with some desired orientations, we then need to compute the set of joint angles that can achieve this desired result. Before we start to compute the inverse kinematics, we will first perform a workspace analysis in MATLAB to get some sense of how our robot will perform in 3D space. Note that in general there is no closed-form solution for inverse kinematics, and solutions are considered on a case-by-case basis. To verify that our forward and inverse kinematics are computed correctly, we will then perform a simple task simulation using MATLAB. The basic idea is that we will predetermine a trajectory for our end effector to follow, and by looking at the animation provided by MATLAB, we can then determine the correctness of our computation. Our original plan was to build our robot using 3D printing and test it in real environment, but unfortunately due to COVID-19 we did not have access to the 3D printer in UCLA. Therefore, we will conduct these tasks when the campus reopens. Detailed explanations and computations for each step will be provided in the following sections.

2 Manipulator Configuration

First, we started the design process by creating a preliminary sketch, shown as in Fig.1. We also assigned frames based on this preliminary sketch, the details of which will be discussed in the following section. As can be seen in the sketch, our robotic arm is composed of four joints: one prismatic and three revolute. The prismatic joint is mounted to a wall near the ceiling and the revolute joint on the platform allows the arm to clean high shelves, the ceiling, and adjacent walls. The two final revolute joints rotate about parallel axes, allowing for the arm to reach away from the wall that it is mounted to. We chose to have one prismatic joint because this allowed us to have maximum reach along the wall without having to make extremely long links, which we would have to do if we chose a four revolute joint configuration. Additionally, we considered having two perpendicular prismatic joints so that the arm could reach any point along the wall. However, we discarded this idea since it would prevent shelves and furniture from

being able to be flush against the mount wall.

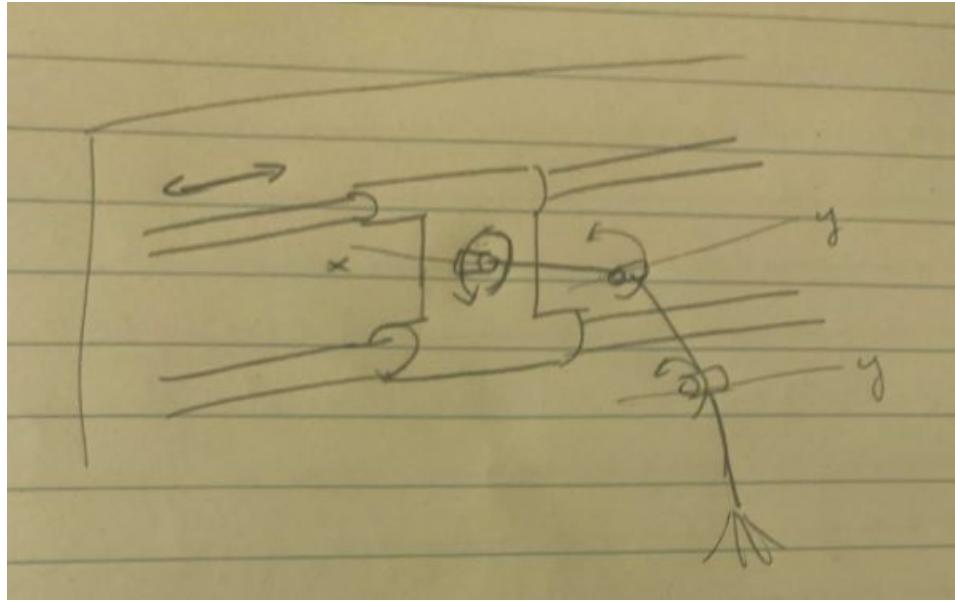


Fig. 1. Preliminary design for the DRUNK

The mechanical principle of the DRUNK robot will be described as follows: For the prismatic joint, we use motor 1 to drive a pulley which attaches to a screw and moves the arm along the prismatic joint. Motor 2 drives the vertical plane of the moving the link 2 to rotate. Motor 3 drives the link 3 to rotate. Similarly, Motor 4 drives the link 4 to rotate. In addition, choosing the length of each link is not an easy task. On the one hand, we want to make the links as long as possible to ensure the reachable work space. On the other hand, the length is limited by the motor torque. The final selection of our link lengths is shown below and the force analysis used will be discussed later in a dedicated section.

Table 1. Length of each links

link	length(m)
d_2	0.38
a_3	0.3
a_4	0.28

To minimize the weight of each link, we will use as few metal materials as possible and use plastic instead. Specifically, as shown in Fig.2, in this DRUNK robot, only the prismatic joint part uses lightweight metal (aluminum alloy), while the rest of the parts (including links, rotation joints, and end effectors) use ABS plastic. There are also multiple different replaceable end effectors like mop, sponge and broom heads which can be used for different tasks and are shown in Fig.3. They all have a standard threaded connector making replacement quite easy.

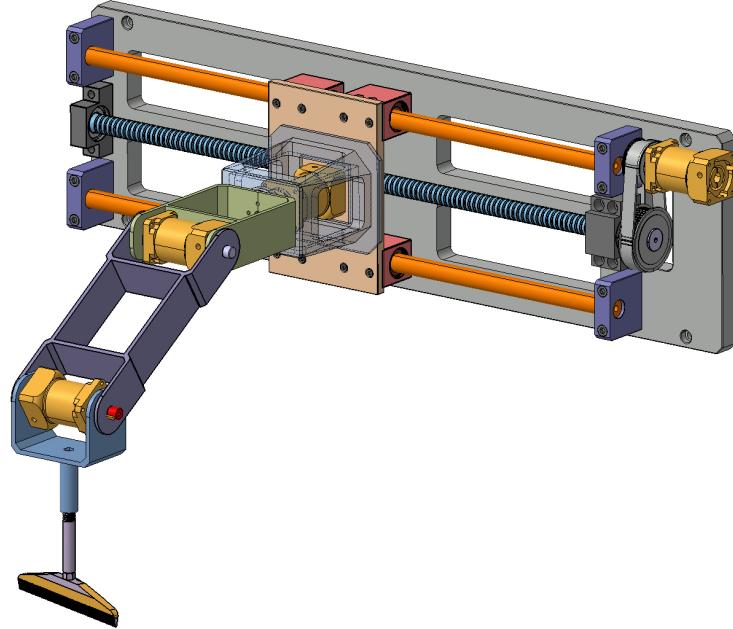


Fig. 2. CAD design for the DRUNK

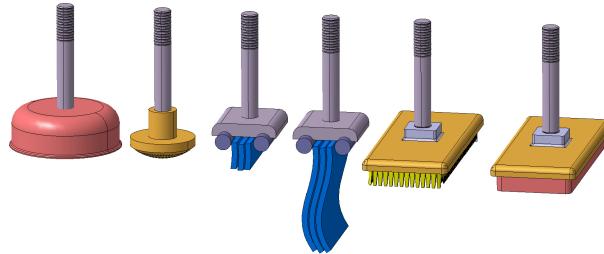


Fig. 3. Multiple replaceable end effectors

3 Forward Kinematics

Before the derivation of kinematics, all link-frames are assigned accordingly as shown in Fig.4.

Then according to Fig.4, the DH parameters table can be obtained easily, which is shown as in Table 2. After obtaining the DH table, using equation (1), the transformation matrix of each frame can be computed .

$${}^{i-1}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

After we obtained the the transformation matrices, we only need to multiply these four matrices together and simplify using some trigonometric identities to get the final result. The final result of the

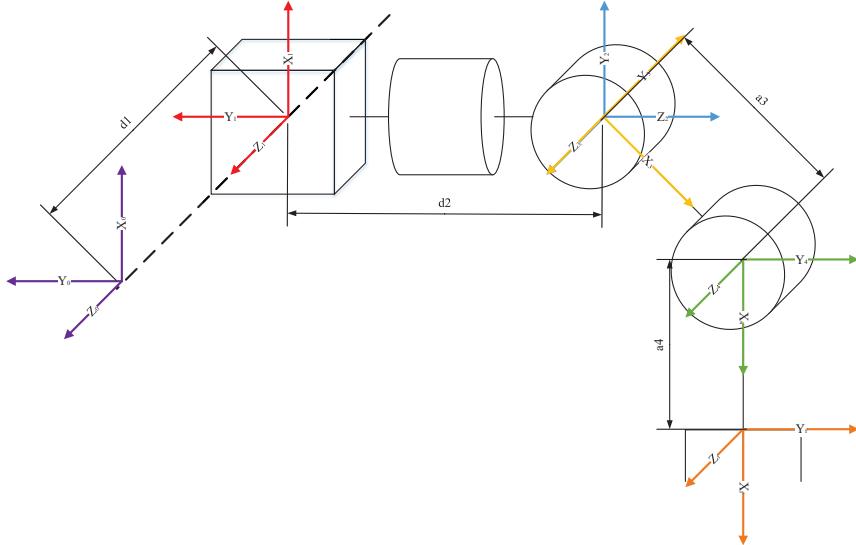


Fig. 4. Link-frame and DH parameters assignments

Table 2. DH Parameters

$i - 1$	i	α_{i-1}	a_{i-1}	d_i	θ_i
0	1	0	0	d_1	0
1	2	90°	0	d_2	θ_2
2	3	-90°	0	0	θ_3
3	4	0	a_3	0	θ_4
4	f	0	a_4	0	0

forward kinematics for our robot is shown in (2).

$${}_f^0T = {}_1^0T {}_2^1T {}_3^2T {}_4^3T {}_f^4T = \begin{bmatrix} c_2c_{34} & -c_2s_{34} & -s_2 & a_4c_2c_{34} + a_3c_2c_3 \\ s_{34} & c_{34} & 0 & a_4s_{34} + a_3s_s - d_2 \\ s_2c_{34} & -s_2s_{34} & c_2 & a_4s_2c_{34} + d_1 + a_3s_2c_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

In order to test the correctness of our calculated Forward Kinematics, we decided to create a simple simulation using MATLAB to demonstrate the movements of each link using different joint angles as inputs. From the above DH table, we can see that the joint variables here are $[d_1, \theta_2, \theta_3, \theta_4]$. Thus, starting from the resting position of our robot, in which all links are pointing in a straight line horizontally, we gradually change each joint variable in the sequence $d_1, \theta_2, \theta_3, \theta_4$. A screenshot of our simulation in MATLAB is provided in Fig. 5.

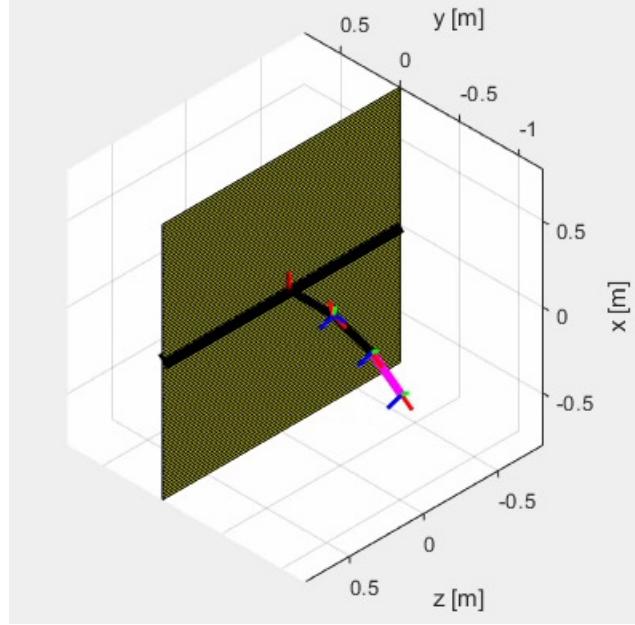


Fig. 5. MATLAB Simulation of Forward Kinematics

4 Inverse Kinematics

Given

$${}_f^0T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_2c_{34} - c_2s_{34} & -s_2 & a_4c_2c_{34} + a_3c_2c_3 \\ s_{34} & c_{34} & a_4s_{34} + a_3s_s - d_2 \\ s_2c_{34} - s_2s_{34} & c_2 & a_4s_2c_{34} + d_1 + a_3s_2c_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

We want to find d_1 , θ_2 , θ_3 , and θ_4 as functions of known values. First we have:

$$\theta_2 = A \tan 2(s_2, c_2) = A \tan 2(-r_{13}, r_{33}) \quad (4)$$

Then we can solve θ_3 by:

$$\begin{aligned} \theta_3 &= A \tan 2[(p_y - a_4s_{34} + d_2)r_{33}, p_x - a_4c_2c_{34}] \\ &= A \tan 2[(p_y - a_4r_{21} + d_2)r_{33}, p_x - a_4r_{33}r_{22}] \end{aligned} \quad (5)$$

Since we have $\theta_3 + \theta_4 = A \tan 2[s_{34}, c_{34}] = A \tan 2[r_{21}, r_{22}]$, θ_4 will be solved.

$$\theta_4 = A \tan 2(r_{21}, r_{22}) - A \tan 2[(p_y - a_4r_{21} + d_2)r_{33}, p_x - a_4r_{33}r_{22}] \quad (6)$$

Given θ_2 , θ_3 , and θ_4 , the last unknown item can be shown with p_z .

$$d_1 = p_z - a_4s_2c_{34} - a_3s_2c_3 = p_z + \frac{p_zr_{13}}{r_{33}} \quad (7)$$

5 Workspace Analysis

In the Workspace Analysis section, we consider two different cleaning modes: with and without limitation. Workspace without limitation mode means we do not set any limitation to the joint angles. As for the workspace with limitation mode, we only allow θ_3 and θ_4 to be in the range $\pm 15^\circ$. Workspace without limitation mode can be used for some general cleaning problems, and workspace with limitation mode mainly focuses on ground sweeping.

5.1 Workspace without limitation

If we fix d_1 , which is the prismatic joint, we will get the spherical reachable work space as shown in Fig. 6(a). Notice that it is a hollow sphere, because the inside part will not be reachable. However, if we allow the robot to move along the slide, the reachable work space will be something like a capsule as shown in Fig. 6(b). This time, all the space inside is reachable. Whether we fix the d_1 or not, there is always no dexterous work space.

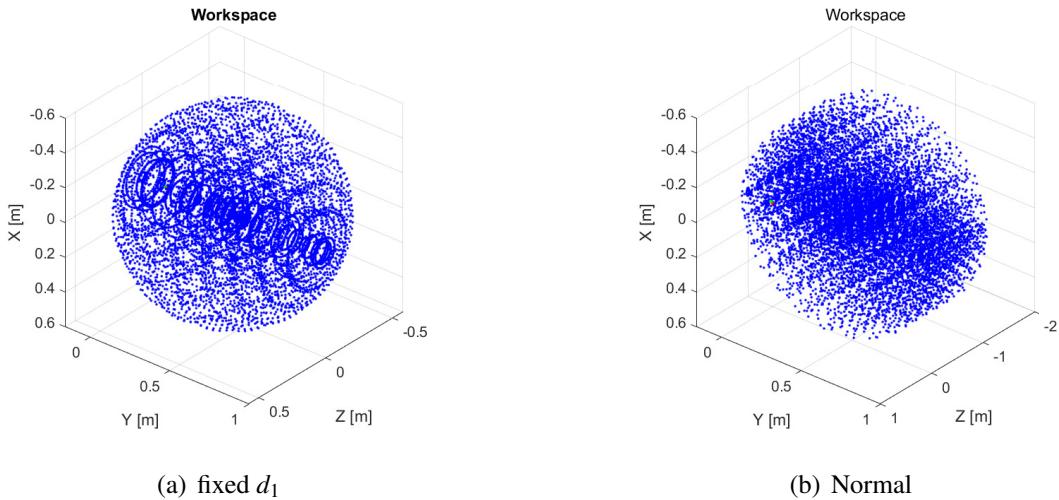


Fig. 6. Work space with limitation

5.2 Workspace with limitation

Similarly, the fixed d_1 and normally reachable work space with limitation will be like Fig. 7(a) and Fig. 7(b).

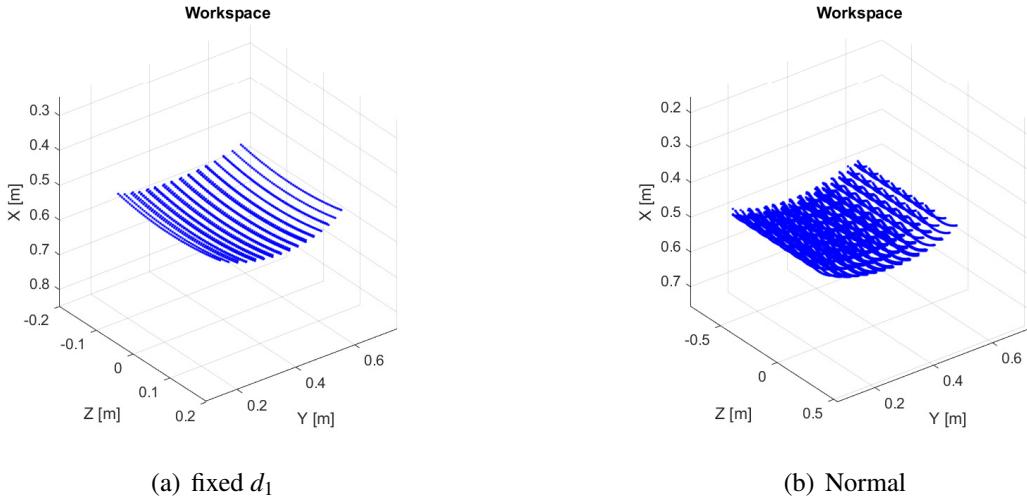


Fig. 7. Work space with limitation,

6 Static Force Analysis

The goal of static force analysis is to determine if the given servo motor, the Dynamixel MX-28AR actuator, is capable of providing enough torque to move the robotic arm throughout its workspace. The Dynamixel actuator has a stall torque of 2.5 N-m. Force analysis was performed in the configuration that placed the most torque on the actuator. This configuration is shown below, which causes the actuator on top of the prismatic joint to experience the most torque.

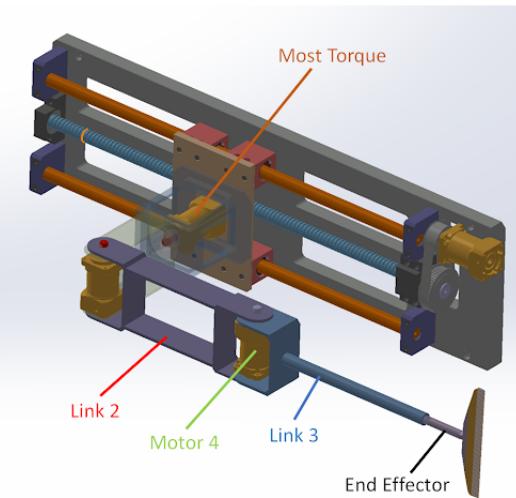


Fig. 8. Static force analysis configuration

As shown in the figure above, the torque on the most-torque actuator is caused by the weight of Link 2, Motor 4, Link 3, and End Effector. If the Dynamixel actuator is capable of providing adequate torque in this configuration, then it will be able to provide enough torque in all other configurations. For each of the torque-contributing components, the moment arm (distance from the center of mass of each component to the rotation axis of the most-torque actuator) was measured in SolidWorks along with the mass of each component. From there, the resultant torque was calculated.

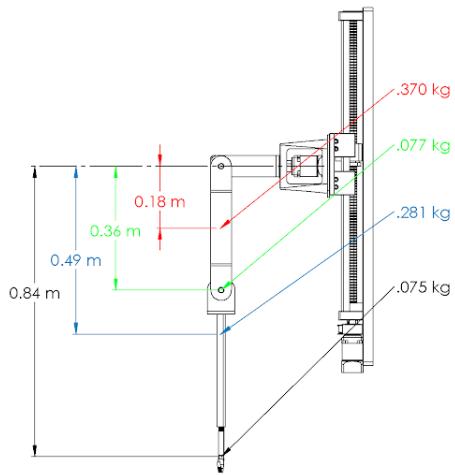


Fig. 9. Original design moment arms and masses

Table 3. Original design torques

Component	Link 2	Motor 4	Link 3	End Effector	Total
Torque (N-m)	0.653	0.272	1.345	0.621	2.891

As shown in the table above, the original design, whose link lengths were arbitrarily set, caused the actuator to experience a torque greater than its stall torque. As a result, the links were redesigned to achieve an adequate maximum torque value.

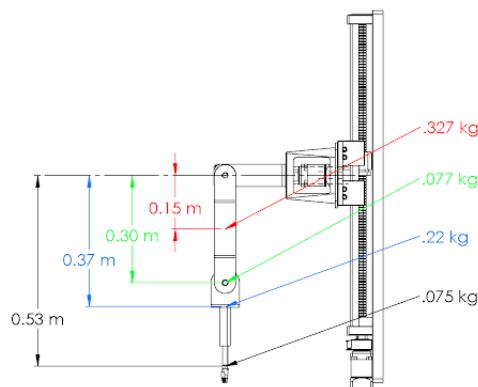


Fig. 10. Revised design moment arms and masses

Table 4. Revised design torques

Component	Link 2	Motor 4	Link 3	End Effector	Total
Torque (N-m)	0.481	0.272	0.792	0.393	1.893

As shown in the figure and table above, the redesigned links are much shorter than the original design. Because of this change, the maximum torque is now 1.893 N-m, which is well below the stall torque of 2.5 N-m.

7 Task Simulation

Using our inverse and forward kinematics in Matlab, we were able to produce several trajectories that would be useful for the DRUNK robot's dusting application. One of those possible trajectories is shown in the figure below, in a back and forth trajectory on a flat plane, similar to the motion one might use to dust off or wipe down a table.

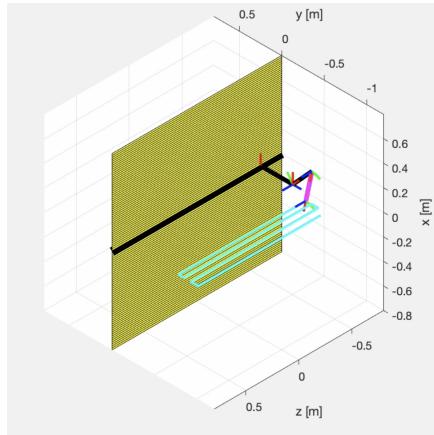


Fig. 11. Possible trajectory for the DRUNK robot's dusting application

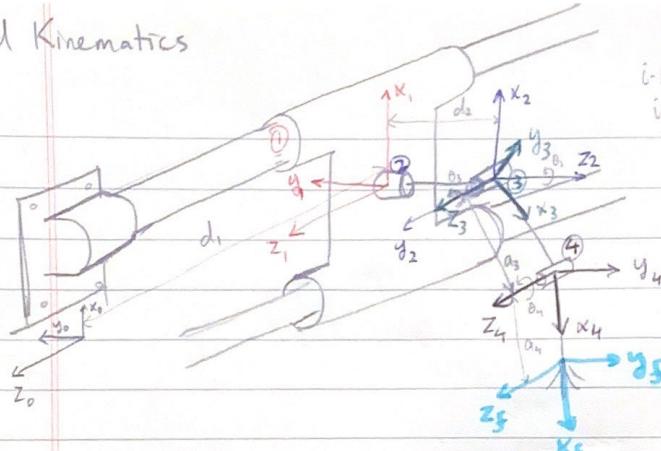
8 Conclusion and Future Works

In this project, we were able to successfully design and simulate a four degree of freedom robotic arm used for wiping down hard-to-reach surfaces. We began with a concept and used forward and inverse kinematics in MATLAB to simulate it. Our design was finalized using CATIA and static force analysis.

In the future, more work can be done to optimize the links connecting the revolute joints to minimize the torque on the base revolute joint motor, allowing the links to be longer and the robotic arm to have a greater reach. In a similar vein, more powerful motors might be chosen to allow the links to be longer and increase the arm's reach. Additionally, we may choose to actually build the robotic arm to see how it performs in real life and to evaluate the accuracy of our simulations and analysis upon campus reopening.

Appendix A: Forward kinematics and inverse kinematics

Forward Kinematics



$${}_{i-1}^i T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_i \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<u>i-1</u>	<u>i</u>	<u>α_{i-1}</u>	<u>a_{i-1}</u>	<u>d_i</u>	<u>θ_i</u>
0	1	0	0	d_1	0
1	2	90°	0	d_2	θ_2
2	3	-90°	0	0	θ_3
3	4	0	a_3	0	θ_4
4	5	0	a_4	0	0

$${}^0 {}^1 T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1 {}^2 T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & -1 & -d_2 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2 {}^3 T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_3 & -c\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^3 {}^4 T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_3 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0 {}^4 T = \begin{bmatrix} c_2 c_{34} & -c_2 s_{34} & -s_2 & a_3 c_2 c_3 \\ s_{34} & c_{34} & 0 & a_3 s_3 - d_2 \\ s_2 c_{34} & -s_2 s_{34} & c_2 & d_1 + a_3 s_2 c_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^4 {}^5 T = \begin{bmatrix} 1 & 0 & 0 & a_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0 {}^5 T = \begin{bmatrix} c_2 c_{34} & -c_2 s_{34} & -s_2 & a_4 c_2 c_{34} + a_3 c_2 c_3 \\ s_{34} & c_{34} & 0 & a_4 s_{34} + a_3 s_3 - d_2 \\ s_2 c_{34} & -s_2 s_{34} & c_2 & a_4 s_2 c_{34} + d_1 + a_3 s_2 c_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 12. Forward Kinematics

Inverse Kinematics

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0_f T = \begin{bmatrix} c_2 c_{34} & -c_2 s_{34} & -s_2 & a_4 c_2 c_{34} + a_3 c_2 c_3 \\ s_{34} & c_{34} & 0 & a_4 s_{34} + a_3 s_3 - d_2 \\ s_2 c_{34} & -s_2 s_{34} & c_2 & a_4 s_2 c_{34} + d_1 + a_3 s_2 c_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Goal ${}^0_f T$

Want to find: $d_1, \theta_2, \theta_3, \theta_4$ as functions of known values

- Known values:
- Goals: $r_{11}, r_{12}, \dots, r_{23}, p_x, p_y, p_z$
 - Link lengths: d_2, a_3, a_4

$$[{}^0_f T(d_1)]^{-1} {}^0_f T = [{}^1_f T(d_1)]^{-1} {}^0_f T = {}^1_f T$$

↓
Goal

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 c_{34} & -c_2 s_{34} & -s_2 & a_4 c_2 c_{34} + a_3 c_2 c_3 \\ s_{34} & c_{34} & 0 & a_4 s_{34} + a_3 s_3 - d_2 \\ s_2 c_{34} & -s_2 s_{34} & c_2 & a_4 s_2 c_{34} + a_3 s_2 c_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z - d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^1_f T = \begin{bmatrix} c_2 c_{34} & -c_2 s_{34} & -s_2 & a_4 c_2 c_{34} + a_3 c_2 c_3 \\ s_{34} & c_{34} & 0 & a_4 s_{34} + a_3 s_3 - d_2 \\ s_2 c_{34} & -s_2 s_{34} & c_2 & a_4 s_2 c_{34} + a_3 s_2 c_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$p_z - d_1 = a_4 s_2 c_{34} + a_3 s_2 c_3$$

$$d_1 = p_z - a_4 s_2 c_{34} - a_3 s_2 c_3 \dots (1)$$

$$r_{23} = 0$$

$$\left. \begin{array}{l} r_{33} = c_2 \\ r_{13} = -s_2 \rightarrow s_2 = -r_{13} \end{array} \right\} \tan \theta_2 = \frac{-r_{13}}{r_{23}} \rightarrow \boxed{\theta_2 = \text{Atan2}(-r_{13}, r_{23})}$$

$$\left. \begin{array}{l} r_{21} = s_{34} \\ r_{22} = c_{34} \end{array} \right\} \tan(\theta_3 + \theta_4) = \frac{r_{21}}{r_{22}} \quad \theta_3 + \theta_4 = \text{Atan2}(r_{21}, r_{22}) \dots (2)$$

$$\rightarrow d_1 = p_z - a_4 (-r_{13}) r_{22} - a_3 (-r_{13}) c_3$$

$$d_1 = p_z + a_4 r_{13} r_{22} + a_3 r_{13} c_3 \dots (3)$$

Need to find θ_3 to get d_1 and θ_4 , then everything will be solved.

Fig. 13. Inverse Kinematics

Appendix B: Blueprint of the DRUNK

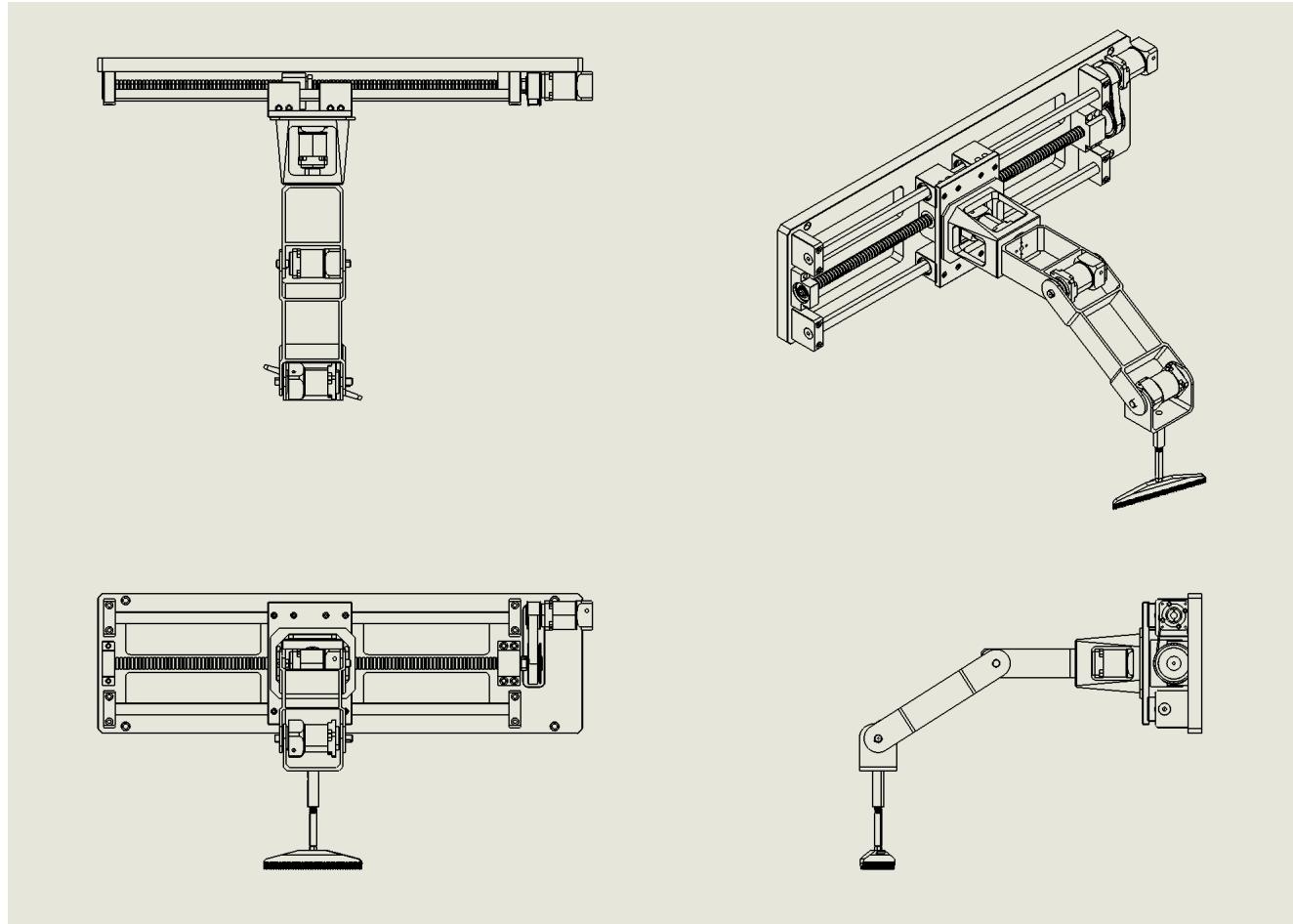


Fig. 14. Inverse Kinematics